

IN THE CLAIMS

What is claimed is:

1. (Currently Amended) A method for constructing a soft device, the method comprising:
 - implementing a software component of the soft device in a virtual machine monitor as a soft device driver, the soft device driver controlling a residual fixed function hardware device represented by a hardware component of the soft device; and
 - making the soft device available for use by one or more virtual machines coupled to the virtual machine monitor, the soft device being independent of any operating system run by the virtual machines.
2. (Cancelled)
3. (Original) The method of claim 1 wherein making the soft device available for use by one or more virtual machines further comprises:
 - exporting an emulation of a fixed function hardware device to said any of the one or more virtual machines.
4. (Original) The method of claim 3 wherein exporting the emulation of the fixed function hardware device comprises:
 - performing computations requested by said any of the one or more virtual machines without notifying a residual fixed function device.
5. (Original) The method of claim 3 wherein exporting the emulation of the fixed function hardware device comprises:
 - transferring an operation requested by said any of the one or more virtual machines to a residual fixed function device; and
 - the residual fixed function device performing the operation requested by said any of the one or more virtual machines.
6. (Original) The method of claim 3 wherein exporting the emulation of the fixed

function hardware device comprises:

performing a portion of computations requested by said any of the one or more virtual machines to a residual fixed function device; and

performing a set of operations on hardware registers of a residual fixed function device to complete a task requested by said any of the one or more virtual machines.

7. (Original) The method of claim 3 wherein exporting the emulation of the fixed function hardware device comprises:

manipulating data stored in memory to effect zero or more transformations; and

transferring data to or from a residual hardware device using a direct memory access (DMA) technique.

8. (Currently Amended) A system comprising:

a hardware platform including a residual fixed function hardware device represented by a hardware component of a soft device;

a virtual machine monitor, coupled to the hardware platform, the virtual machine monitor including a driver of the soft device, the soft device driver controlling the residual fixed function hardware device; and

one or more virtual machines, coupled to the virtual machine monitor, the one or more virtual machines utilizing the soft device when needed, the soft device being independent of any operating system run by the virtual machines.

9. (Cancelled)

10. (Original) The system of claim 8 wherein the driver of the soft device is to export an emulation of a fixed function hardware device to said any of the one or more virtual machines.

11. (Currently Amended) A method for constructing a soft device, the method comprising:

implementing a software component of the soft device in a first virtual machine, the software component controlling a residual fixed function hardware device represented by a

hardware component of the soft device; and

making the soft device available for use by a second virtual machine, the soft device being independent of an operating system run by the second virtual machines.

12. (Cancelled)

13. (Original) The method of claim 11 wherein making the soft device available for use by the second virtual machine further comprises:

presenting the first virtual machine to the second virtual machine as an external device; and

emulating communication between the first virtual machine and the second virtual machine.

14. (Original) The method of claim 13 wherein emulating communication further comprises:

providing a virtualized serial communications link;

providing a virtualized serial communications port to each of the first virtual machine and the second virtual machine;

linking the virtualized serial communications port provided to the first virtual machine to the software component of the soft device using reflection software;

trapping each access by one of the first virtual machine and the second virtual machine to the virtualized serial communication port; and

reflecting said each access to the other of the first virtual machine and the second virtual machine via the virtualized serial communications link.

15. (Original) The method of claim 13 wherein emulating communication further comprises:

providing a virtualized universal serial bus (USB) to USB bridge device;

providing a virtualized USB host controller to each of the first virtual machine and the second virtual machine;

linking the virtualized USB host controller provided to the first virtual machine to the software component of the soft device using reflection software;

trapping each access by one of the first virtual machine and the second virtual machine to the virtualized USB host controller; and

reflecting said each access to the other of the first virtual machine and the second virtual machine via the virtualized USB to USB bridge device.

16. (Original) The method of claim 11 wherein making the soft device available for use by the second virtual machine further comprises:

presenting the first virtual machine to the second virtual machine as an internal device; and

emulating communication between the first virtual machine and the second virtual machine.

17. (Original) The method of claim 16 wherein emulating communication further comprises:

providing a virtualized peripheral component interconnect (PCI) bus;

linking the virtualized PCI bus to the software component of the soft device using reflection software;

trapping each access by one of the first virtual machine and the second virtual machine to the virtualized PCI bus; and

reflecting said each access to the other of the first virtual machine and the second virtual machine.

18. (Original) The method of claim 11 wherein making the soft device available for use by the second virtual machine further comprises:

emulating a network communication between the first virtual machine and the second virtual machine by providing a virtual network interface card (NIC) to each of the first virtual machine and the second virtual machine.

19. (Original) The method of claim 11 wherein making the soft device available for use by the second virtual machine further comprises:

presenting the first virtual machine to the second virtual machine as a hardware device; and

emulating communication between the first virtual machine and the second virtual machine.

20. (Original) The method of claim 19 wherein emulating communication further comprises:

- providing a virtualized peripheral component interconnect (PCI) bus;
- trapping each access by one of the first virtual machine and the second virtual machine to the virtualized PCI bus; and
- reflecting said each access to the other of the first virtual machine and the second virtual machine via the virtualized PCI bus.

21. (Original) The method of claim 19 wherein emulating communication further comprises:

- providing a virtualized universal serial bus (USB) connection;
- providing a virtualized USB bus interface to the first virtual machine;
- providing a virtualized USB host controller to the second virtual machine;
- trapping each access by one of the first virtual machine and the second virtual machine to the virtualized USB bus; and
- reflecting said each access to the other of the first virtual machine and the second virtual machine via the virtualized USB connection.

22. (Original) The method of claim 19 wherein the hardware device is any one of a PCI card, an external USB device, an internal USB device, and any other standard personal computer peripheral device.

23. (Original) The method of claim 19 wherein presenting the first virtual machine to the second virtual machine as a hardware device further comprises:

- configuring the first virtual machine to match the hardware device.

24. (Original) The method of claim 23 wherein the software component of the soft device comprises at least a portion of software of a fixed function device.

25. (Original) The method of claim 24 further comprising:
varying the portion of software that is used as the software component depending on how closely the first virtual machine matches the hardware device.
26. (Currently Amended) A method for constructing a soft device, the method comprising:
implementing software components of the soft device in a plurality of dedicated virtual machines, each of the software components controlling a residual fixed function hardware device represented by a hardware component of the soft device; and
making the soft device available for use by a main virtual machine, the soft device independent of an operating system run by the main virtual machine.
27. (Cancelled)
28. (Original) The method of claim 26 wherein making the soft device available for use by the main virtual machine further comprises:
presenting the plurality of dedicated virtual machines to the main virtual machine as a hardware device; and
emulating communication between the plurality of dedicated virtual machines and between each of the plurality of dedicated virtual machines and the main virtual machine.
29. (Original) The method of claim 28 wherein emulating communication further comprises:
providing a virtualized communication means to the plurality of dedicated virtual machines and to the main virtual machine.
30. (Original) The method of claim 29 wherein the virtualized communication means is any one of a virtualized serial communications link, a virtualized universal serial bus (USB) to USB bridge device, a virtualized peripheral component interconnect (PCI) bus, a virtual network interface card, and a virtualized USB connection.
31. (Currently Amended) A system comprising:

a hardware platform including a residual fixed function hardware device represented by a hardware component of a soft device;
a virtual machine monitor, coupled to the hardware platform; and
a plurality of virtual machines, coupled to the virtual machine monitor, the plurality of virtual machines including one or more dedicated virtual machines with one or more software components of the soft device implemented therein, and remaining one or more virtual machines utilizing the soft device when needed, the soft device being independent of any operating system run by the virtual machines.

32. (Cancelled)

33. (Original) The system of claim 31 wherein the VMM is to make the soft device available by presenting the dedicated virtual machines to the remaining virtual machines as one or more external devices and emulating communication between the plurality of virtual machines.

34. (Original) The system of claim 31 wherein the VMM is to make the soft device available by presenting the dedicated virtual machines to the remaining virtual machines as one or more internal devices and emulating communication between the plurality of virtual machines.

35. (Original) The system of claim 31 wherein the VMM is to make the soft device available by emulating a network communication between the plurality of virtual machines by providing a virtual network interface card (NIC) to each of the plurality of virtual machines.

36. (Original) The system of claim 31 wherein the VMM is to make the soft device available by presenting the dedicated virtual machines to the remaining virtual machines as a hardware device and emulating communication between the plurality of virtual machines.

37. (Original) The system of claim 36 wherein the hardware device is any one of a PCI card, an external USB device, an internal USB device, and any other standard personal

computer peripheral device.

38. (Original) The system of claim 36 wherein the VMM is to emulate communication by providing a virtualized communication means to the plurality of virtual machines.

39. (Original) The system of claim 38 wherein the virtualized communication means is any one of a virtualized serial communications link, a virtualized universal serial bus (USB) to USB bridge device, a virtualized peripheral component interconnect (PCI) bus, a virtual network interface card, and a virtualized USB connection.

40. (Currently Amended) A computer readable medium that provides instructions, which when executed on a processor, cause said processor to perform operations comprising:

implementing a software component of a soft device in a virtual machine monitor as a soft device driver, the soft device driver controlling a residual fixed function hardware device represented by a hardware component of the soft device; and

making the soft device available for use by one or more virtual machines coupled to the virtual machine monitor, the soft device being independent of any operating system run by the virtual machines.

41. (Cancelled)

42. (Currently Amended) A computer readable medium that provides instructions, which when executed on a processor, cause said processor to perform operations comprising:

implementing a software component of the soft device in a first virtual machine, the software component controlling a residual fixed function hardware device represented by a hardware component of the soft device; and

making the soft device available for use by a second virtual machine, the soft device being independent of an operating system run by the second virtual machine.

43. (Cancelled)

44. (Currently Amended) A computer readable medium that provides instructions, which

when executed on a processor, cause said processor to perform operations comprising:

implementing software components of the soft device in a plurality of dedicated virtual machines, each of the software components controlling a residual fixed function hardware device represented by a hardware component of the soft device; and

making the soft device available for use by a main virtual machine, the soft device being independent of an operating system run by the main virtual machine.

45. (Cancelled)